**How do you update the records with or without using Update Strategy?**

We can use the session configurations to update the records. We can have several options for handling database operations such as insert, update, delete.

During session configuration, you can select a single database operation for all rows using the Treat Source Rows As setting from the ‘Properties’ tab of the session.

- **Insert: –** Treat all rows as inserts.
- **Delete: –** Treat all rows as deletes.
- **Update: –** Treat all rows as updates.
- **Data Driven :-** Integration Service follows instructions coded into Update Strategy flag rows for insert, delete, update, or reject.

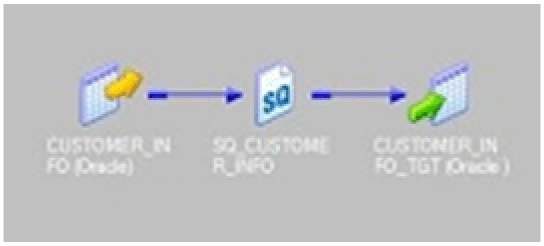
Once determined how to treat all rows in the session, we can also set options for individual rows, which gives additional control over how each rows behaves. We need to define these options in the Transformations view on mapping tab of the session properties.

- **Insert: –** Select this option to insert a row into a target table.
- **Delete: –** Select this option to delete a row from a table.
- **Update :-** You have the following options in this situation:
    - Update as Update: – Update each row flagged for update if it exists in the target table.
    - Update as Insert: – Insert each row flagged for update.
    - Update else Insert: – Update the row if it exists. Otherwise, insert it.
- **Truncate Table: –** Select this option to truncate the target table before loading data.

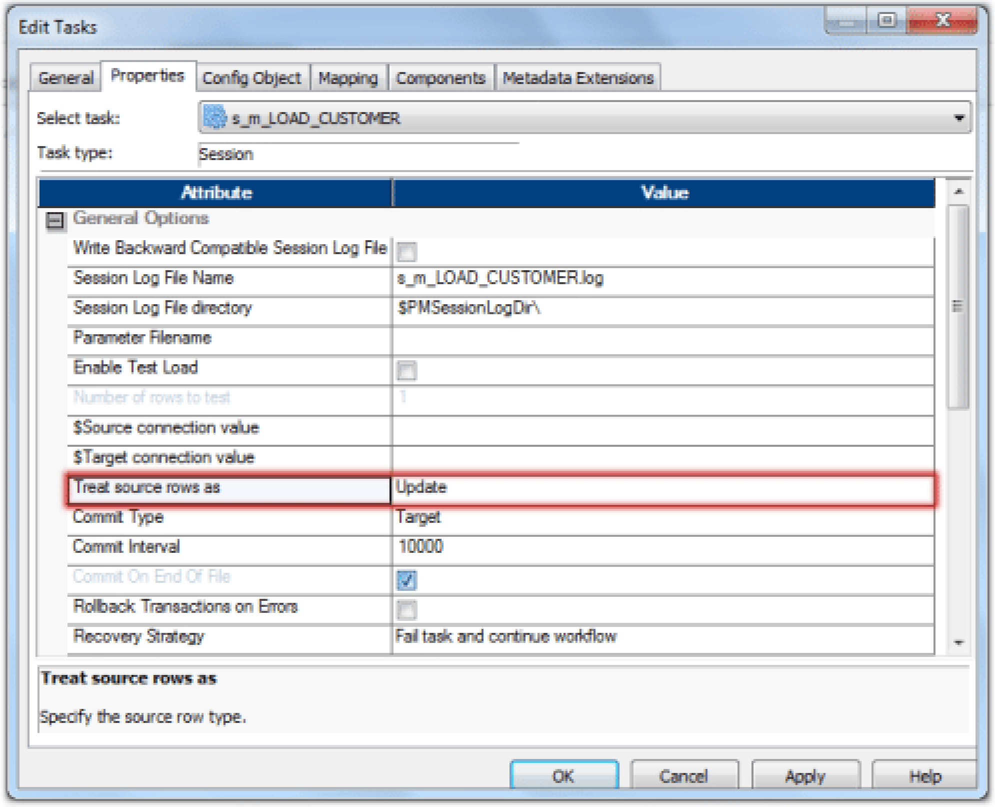
Steps:

1. Design the mapping just like an ‘INSERT’ only mapping, without Lookup, Update

Strategy Transformation.



2. First set Treat Source Rows As property as shown in below image.



Select task: s_m_LOAD_CUSTOMER

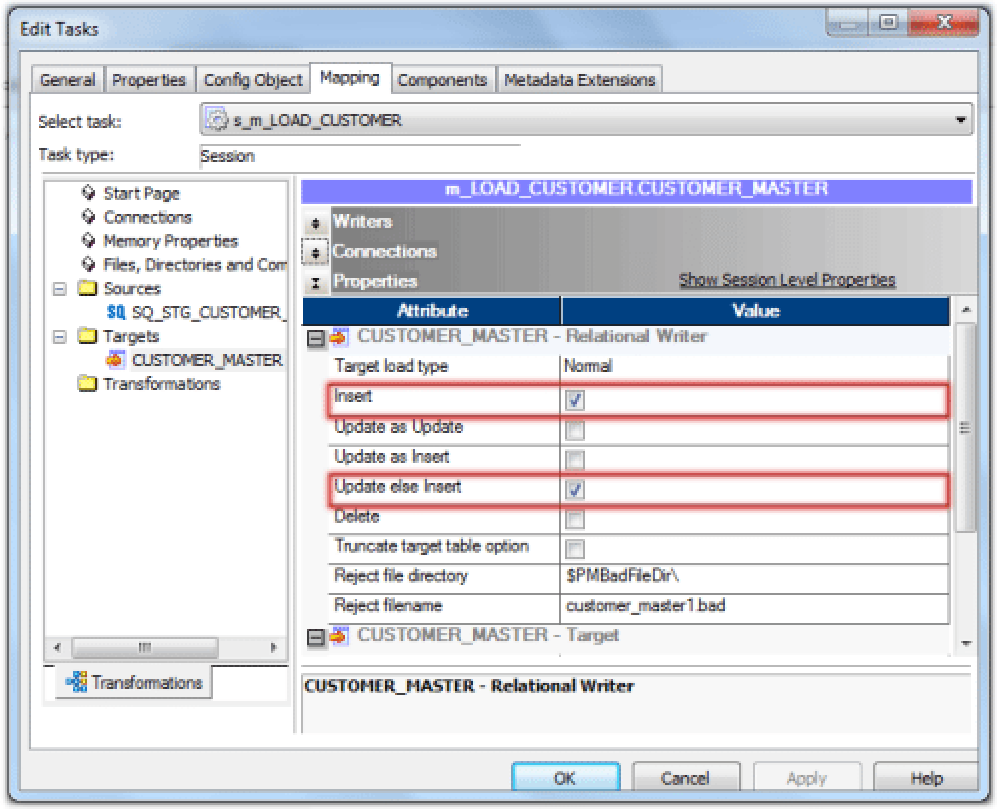
Task type: Session

| Attribute | Value |
|---|---|
| General Options | |
| Write Backward Compatible Session Log File | □ |
| Session Log File Name | s_m_LOAD_CUSTOMER.log |
| Session Log File directory | $PMSessionLogDir\ |
| Parameter Filename | |
| Enable Test Load | □ |
| Number of rows to test | 1 |
| $Source connection value | |
| $Target connection value | |
| Treat source rows as | Update |
| Commit Type | Target |
| Commit Interval | 10000 |
| Commit On End Of File | ☑ |
| Rollback Transactions on Errors | □ |
| Recovery Strategy | Fail task and continue workflow |

**Treat source rows as**

Specify the source row type.

3. Next, set the properties for the target table as shown below. Choose the properties Insert and Update else Insert.



Select task: s_m_LOAD_CUSTOMER

Task type: Session

m_LOAD_CUSTOMER.CUSTOMER_MASTER

| Attribute | Value |
|---|---|
| CUSTOMER_MASTER - Relational Writer | |
| Target load type | Normal |
| Insert | ☑ |
| Update as Update | □ |
| Update as Insert | □ |
| Update else Insert | ☑ |
| Delete | □ |
| Truncate target table option | □ |
| Reject file directory | $PMBadFileDir\ |
| Reject filename | customer_master1.bad |
| CUSTOMER_MASTER - Target | |

CUSTOMER_MASTER - Relational Writer

These options will make the session as Update and Insert records without using Update Strategy in Target Table.

When we need to update a huge table with few records and less inserts, we can use this solution to improve the session performance.

The solutions for such situations is not to use Lookup Transformation and Update Strategy to insert and update records.

The Lookup Transformation may not perform better as the lookup table size increases and it also degrades the performance.

**How do you load unique records into one target table and duplicate records into a different target table?**

**Source Table:**

| COL1 | COL2 | COL3 |
|---|---|---|
| a | b | c |
| x | y | z |
| a | b | c |
| r | f | u |
| a | b | c |
| v | f | r |
| v | f | r |

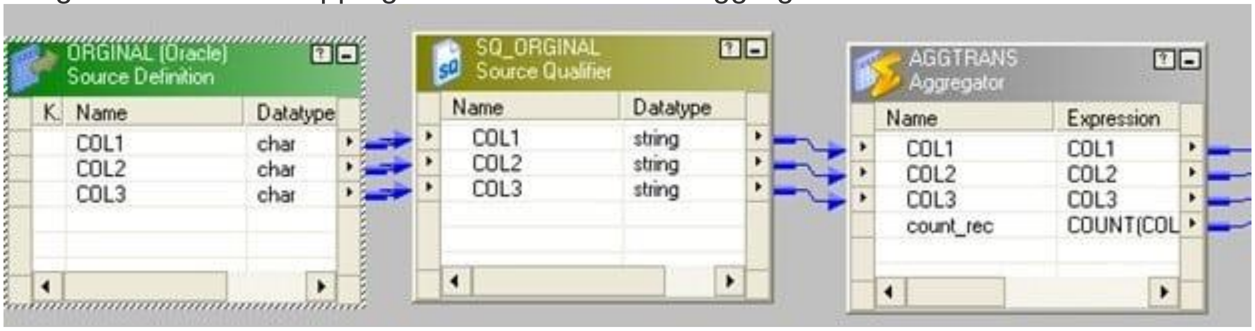
**Target Table 1**: Table containing all the unique rows

| COL1 | COL2 | COL3 |
|---|---|---|
| a | b | c |
| x | y | z |
| r | f | u |
| v | f | r |

**Target Table 2:** Table containing all the duplicate rows

| COL1 | COL2 | COL3 |
|---|---|---|
| a | b | c |
| a | b | c |
| v | f | r |

1. Drag the source to mapping and connect it to an aggregator transformation.



ORGINAL (Oracle)
Source Definition

| K. | Name | Datatype |
|---|---|---|
| | COL1 | char |
| | COL2 | char |
| | COL3 | char |

SQ_ORGINAL
Source Qualifier

| Name | Datatype |
|---|---|
| COL1 | string |
| COL2 | string |
| COL3 | string |

AGGTRANS
Aggregator

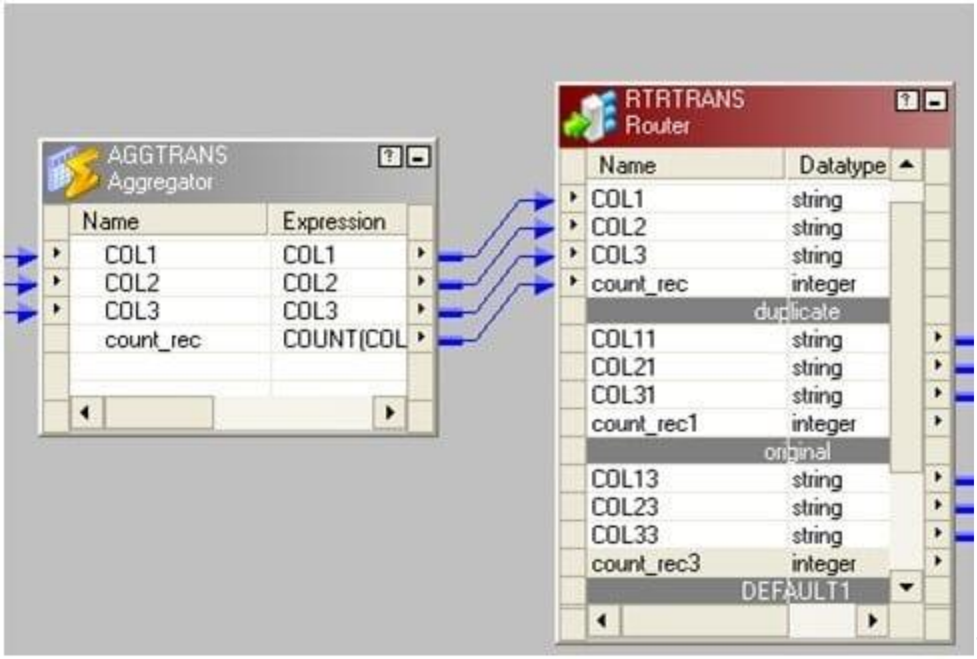
| Name | Expression |
|---|---|
| COL1 | COL1 |
| COL2 | COL2 |
| COL3 | COL3 |
| count_rec | COUNT(COL |

2. In aggregator transformation, group by the key column and add a new port. Call it count_rec to count  the key column.

Connect  a router to the aggregator from the previous step. In router make two groups:

one named “original” and another as “duplicate”.

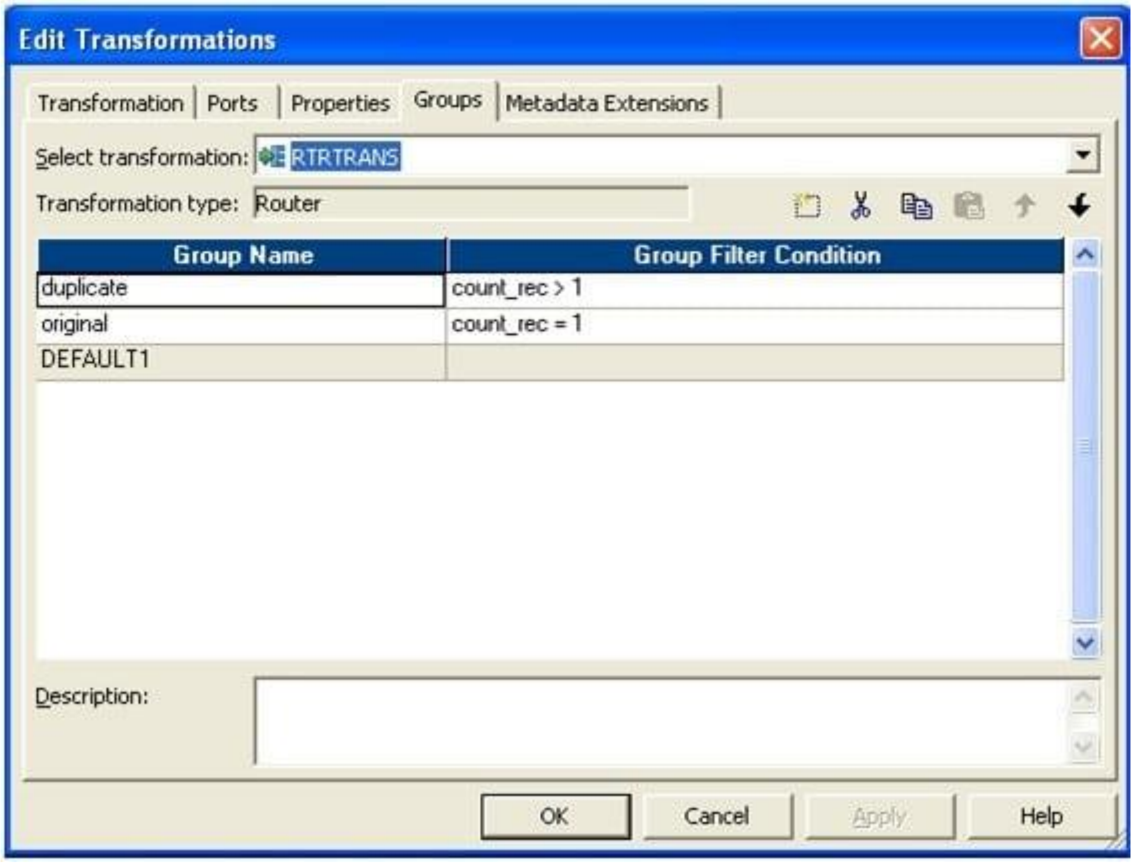
In original write count_rec=1 and in duplicate write count_rec>1.



| Name | Expression |
|---|---|
| COL1 | COL1 |
| COL2 | COL2 |
| COL3 | COL3 |
| count_rec | COUNT(COL |

| Name | Datatype |
|---|---|
| COL1 | string |
| COL2 | string |
| COL3 | string |
| count_rec | integer |
| duplicate | |
| COL11 | string |
| COL21 | string |
| COL31 | string |
| count_rec1 | integer |
| original | |
| COL13 | string |
| COL23 | string |
| COL33 | string |
| count_rec3 | integer |
| DEFAULT1 | |

The picture below depicts the group name and the filter conditions.



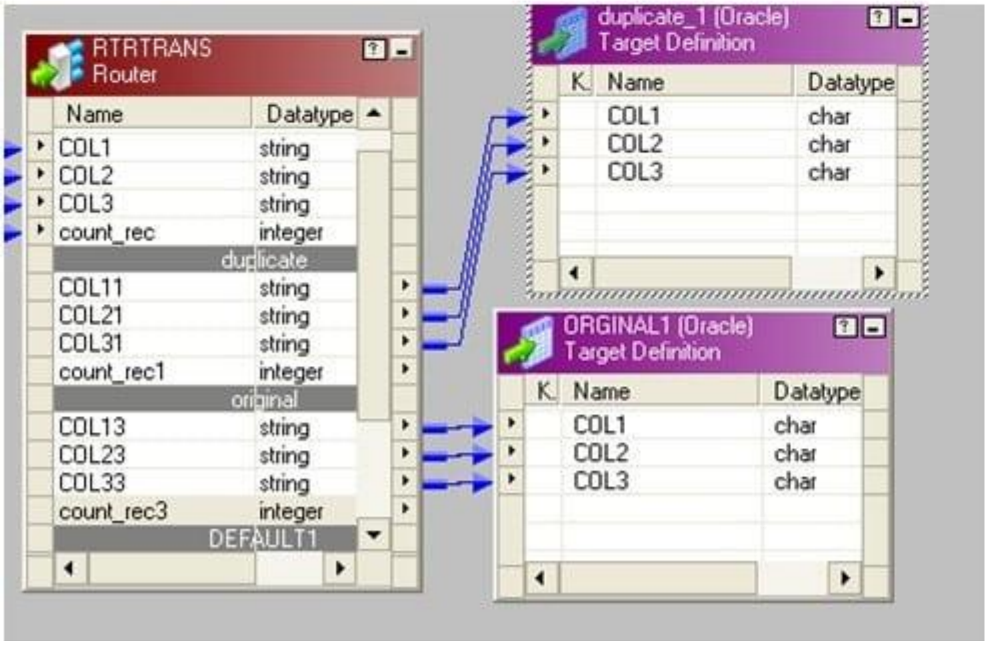
Select transformation: RTRTRANS

Transformation type: Router

| Group Name | Group Filter Condition |
|---|---|
| duplicate | count_rec > 1 |
| original | count_rec = 1 |
| DEFAULT1 | |

Description:

Connect two groups to corresponding target tables.



RTRTRANS
Router

| Name | Datatype |
|---|---|
| COL1 | string |
| COL2 | string |
| COL3 | string |
| count_rec | integer |
| duplicate | |
| COL11 | string |
| COL21 | string |
| COL31 | string |
| count_rec1 | integer |
| original | |
| COL13 | string |
| COL23 | string |
| COL33 | string |
| count_rec3 | integer |
| DEFAULT1 | |

duplicate_1 (Oracle)
Target Definition

| K. | Name | Datatype |
|---|---|---|
| | COL1 | char |
| | COL2 | char |
| | COL3 | char |

ORGINAL1 (Oracle)
Target Definition

| K. | Name | Datatype |
|---|---|---|
| | COL1 | char |
| | COL2 | char |
| | COL3 | char |